

Informazioni sull'esame

L'esame scritto avverrà secondo le indicazioni dell'Ateneo, usando la piattaforma EOL ([eol.unibo.it](http://eol.unibo.it)) e il sistema di videoconferenza Zoom. Tuttavia, durante la prova il docente o un suo assistente chiederà a turno a tutti i partecipanti di spostarsi su una piattaforma diversa (probabilmente Google Docs) per una domanda orale sugli stessi contenuti dello scritto. Lo studente dovrà dunque collegarsi sulla nuova piattaforma, rispondere alla domanda SIA ad alta voce SIA scrivendo la risposta e motivando il ragionamento che sta dietro alla risposta fornita. Al termine di questa singola domanda potrà tornare a completare l'esame scritto su EOL.

Informazioni sul progetto

Ogni informazione sul progetto riporterà sempre i destinatari (tutti, 9CFU+6CFUold, solo 9CFU) e la data a partire dalla quale l'informazione diventerà vincolante (e.g. diventa un requisito obbligatorio per il progetto)

**Accessibilità (9CFU e 6CFUold):** questi sono ulteriori requisiti di progetto per quanto riguarda la realizzazione di applicazioni accessibili. Sono requisiti opzionali per i 6 CFU e obbligatorie per i 9 CFU e i 6CFUold.

- Almeno una delle storie presentate deve essere accessibile (giocabile senza incontrare barriere) anche da persone non vedenti. Gli adattamenti effettuati non dovrebbero rendere il gioco più facile da giocare né meno interessante, ma solo giocabile (usare i principi WCAG 2.1 come ispirazione)!
- Lo strumento di creazione e, di conseguenza, i componenti lato server, devono essere implementati per consentire l'adozione degli accorgimenti necessari (ad es. descrizioni testuali delle immagini, contesto più approfondito per fornire informazioni altrimenti disponibili solo attraverso la vista, ecc.).
- L'interfaccia del player deve essere accessibile e pienamente utilizzabile mediante screen reader, utilizzare WCAG 2.1 livello AA come riferimento. Non effettueremo una valutazione dell'accessibilità come richiesto dalle leggi e dagli standard internazionali, ma l'interfaccia non dovrebbe presentare problemi che ne impediscano l'uso da parte dei non vedenti, né dovrebbe renderne più difficile l'utilizzo da parte loro.
- Il player deve indicare chiaramente quando una storia è accessibile, e per quali disabilità: opzionalmente è possibile creare storie giocabili anche da utenti con altre disabilità.
- Rendere accessibile l'interfaccia della piattaforma di authoring è completamente facoltativo, ma non esitate a comunicarci se ci avete lavorato. Essa deve però consentire (ma non richiedere obbligatoriamente) agli utenti di fornire tutte le informazioni necessarie per rendere accessibile una storia (ad es. descrizioni di testi per immagini, attività alternative, ecc.).

F.A.Q.

**Q:** *Quando l'utente crea la storia deve poter scegliere fra dei widget predisposti da noi oppure li può anche inserire lui(ad esempio con dei JS+CSS personali)? Nel caso potesse farlo, quanta libertà gli diamo?*

**A:** Voi dovete solo fornire dei widget e dei CSS di default, e lasciare al creatore delle storie la possibilità di creare dei propri stili e widget alternativi, e dovete fornire nel JSON della storia il modo di specificarli. e di permettere a questa o quella schermata della storia di usarli. Libertà: tutta quella che è possibile fornire senza richiedere programmazione da parte del creatore di storie. Tuttavia nessun CSS o widget personale deve essere

indispensabile: in assenza di CSS o widget personali la storia sarà meno bella, meno personale, meno creativa, ma perfettamente funzionante.

**Q:** *Dato che ogni attività ha la possibilità di portare ad attività diverse in base alla risposta, dobbiamo permettere anche che una missione porti a missioni diverse in base alle attività svolte o possiamo considerare che le attività di una missione portino sempre allo stesso risultato?*

**A:** Una storia è composta di missioni che sono composte di attività. Le missioni POSSONO essere parallele (ad esempio la mia squadra fa missione 1, 2 e 3, la tua fa 3, 2 4, ecc.) Questo significa che la sequenza delle missioni deve poter essere decisa in qualche maniera (ad esempio attraverso una proprietà nextMission statica oppure una funzione nextMission che sulla base dell'andamento della storia decide quale sarà la prossima missione (e quindi porta alla prima attività di quella missione).

**Q:** *E' possibile che per lo stesso gruppo ci siano missioni parallele, ad esempio la missione 2 è disponibile senza finire la missione 1 o sono per forza sequenziali? stessa domanda per le attività*

**A:** Mi sembra una complicazione inutile. Il gruppo esegue un'attività alla volta. e l'attività appartiene ad una missione. Se avessi due missioni attive contemporaneamente, vorrebbe dire che la stessa attività appartiene a due missioni, oppure che ho due attività da svolgere allo stesso tempo? Diventerebbe troppo complicato.

**Q:** *negli esami per la parte di html+css è possibile aggiungere class o ID agli attributi html?*

**A:** di solito gli esercizi dicono se è possibile aggiungere o meno class e id. Se non lo escludono esplicitamente, potete assumere che sia possibile farlo. Ma a volte, per pura crudeltà, lo impediamo apposta.

**Q:** *ci chiedevamo se jquery ui fosse consentito come framework (ricordo che aveva detto che jquery, angular, vue, react e node fossero approvati di default, ma preferiamo essere sicuri che anche i framework "collegati" lo siano);*

**A:** jquery UI è assolutamente consentita. In generale tutte le librerie Javascript pure sono consentite, e sono proibite solo quelle che impongono una compilazione del codice sorgente.

**Q:** *per l'architettura dei giochi è obbligatorio caricare tutto il contenuto nella pagina con il famoso json o possiamo utilizzare un servizio rest che restituisce pagine o frammenti html in base allo svolgimento e al json di configurazione? Perché nel caso del caricamento unico abbiamo pensato che l'unica soluzione sia creare gli elementi da aggiungere al dom con jquery, ma mi è sembrato un po' macchinoso come processo, inoltre avere delle pagine agevola la fase di test del layout;*

**A:** Il progetto da 6 CFU assume che non vi sia logica server-side, e quindi tutte le risorse necessarie per svolgere il gioco debbono essere caricate in partenza e vengono solo visualizzate o nascoste durante la partita a seconda delle azioni del giocatore. Il progetto da 9CFU è più libero da questo punto di vista, perché prevede una certa quantità di logica server-side a disposizione per cose complesse. Tuttavia questa NON DEVE essere una scusa per scaricare l'intera logica applicativa sul server e fare del client un'applicazione banale di caricamento e visualizzazione di contenuti predeterminati (equivalenti a navigazione tra risorsa HTML e risorsa HTML). Aggiungo: in un setting realistico come un museo in un palazzo antico, può tranquillamente succedere che il WiFi non sia presente in ogni ambiente o non abbia la banda sufficiente per tutti i giocatori. Per questo motivo

possiamo ipotizzare che l'atrio del museo sia ben coperto dalla rete (ovvero l'ambiente dove inizia il gioco e da cui il giocatore scarica le risorse necessarie per giocare) e poi può giocare tranquillamente senza preoccuparsi della copertura di rete. Infine: è molto semplice scivolare da queste premesse in un'applicazione web tradizionale in cui il browser visualizza solo risorse statiche e tutta la logica applicativa (e.g., stato della partita, scelta della prossima attività, caricamento degli indizi ecc.) sia mantenuto server-side come si faceva nel 1998. Questo non va bene e tenderei ad escluderlo. Limiterei invece appunto i contenuti extra ai soli contenuti "speciali" non appartenenti alla partita "normale", ad esempio informazioni temporanee che il docente o il curatore del museo può fornire in maniera imprevedibile (ad esempio: "attenzione: grande assembramento nella sala III, evitatela se potete") o per aiutare un gruppo in difficoltà (ad esempio: "Se proprio non riuscite a trovare l'indizio che vi serve, cercate vicino ad un vaso simile a questo: <IMG>")

**Q:** *ha senso salvare dei frammenti html in dei file e caricarli dinamicamente con ajax? In caso di risposta affermativa: qual è l'architettura maggiormente usata per farlo (db relazionale, file html...)?*

**A:** Questo va scritto in maniera chiara ed esplicita: in NESSUN CASO l'interazione tra server e client avviene attraverso lo scambio di frammenti HTML. Le informazioni che il server passa al client, sia in partenza sia via Ajax, sono dati astratti senza visualizzazione, in formato JSON. HTML non è un formato accettabile di scambio di informazioni mai.

**Abis:** Il meccanismo di storage server-side delle informazioni è a vostra libera scelta, ma non vorrei che scadeste automaticamente in ciò che già conoscete. In particolare, database sì o no non è una scelta rilevante: il client accede ad un URI, e questo URI è quello di un'API REST che specifica gli oggetti da caricare e non la natura o l'identità dell'applicazione server-side che li fornisce. Che dietro ci sia un database relazionale, un database NOSQL o una directory di file statici, è totalmente invisibile al client, e non deve essere MAI determinante per il funzionamento del client. Infatti personalmente ritengo che non ci sia nessuna necessità di un database in tutto il progetto (pochi dati, nessuna autenticazione, nessuna persistenza nel tempo delle partite giocate, ecc.).

**Q:** *avere layout graficamente non coerenti sarebbe penalizzante? Mi piacerebbe fare alcuni widget in puro css, ma ovviamente non è possibile per tutto... alcune parti dei widget può essere interessante costruirle sfruttando esclusivamente css per fare cose graficamente gradevoli, altri richiedono la combinazione di immagini (o almeno io così lo immaginavo)...penso ad esempio al taccuino che è un'immagine, mentre una calcolatrice la costruisco benissimo con solo css: ovviamente non saranno mai belli da visualizzare insieme, però il tipo di "esercizio" è a mio parere differente.*

**A:** sicuramente potete e dovete pensare a parti dell'applicazione che non soltanto accendono o spengono del CSS, ma cambiano anche frammenti del DOM visualizzando o nascondendo strutture, testi o immagini. A vostro piacimento decidere come.

**Q:** *Nell'editor DEVE essere possibile modificare direttamente CSS oppure si può fare indirettamente tramite un widget che chieda di inserire alcune personalizzazioni di stile, poi settate concretamente dal programma?*

**A:** Qualunque meccanismo di modifica del CSS è accettabile, anche l'inclusione di codice fatto da altri, purché sia integrato nell'applicazione (cioè non richieda di andare su un sito web terzo). Ah, e quando dico "codice fatto da altri" sono esclusi evidentemente i progetti di vostri compagni di corso.

**Q:** Possiamo usare material UI (libreria di componenti react "prefabbricati")?

**A:** Potete usare componenti react prefabbricati senza problemi

**Q:** Data la necessaria customizzabilità delle attività e invece unica interfaccia degli ambienti editor e valutatore, va bene (da un punto di vista semantico, non tecnico) usare soluzioni grafiche diverse per player rispetto a valutatore+editor?

**A:** Player e editor sono ambienti diversi. Se sono diversi graficamente non è un problema

**Q:** Cosa intende (nella slide 34) con la frase "almeno un una (storia) per ogni tipo di gioco e per ogni fascia di età"

**Q:** In tutto gli esempi devono essere 9 vero? 3 fasce di età diverse per ogni gruppo ( che sono 3 gruppi ).

**Q:** È corretto creare un tipo di gioco (individuale, piccoli gruppi, classe) per ogni fascia d'età, oppure dobbiamo creare tutti i tipi di gioco per ciascuna fascia d'età? (9 in tutto).

**A:** Questa è una cosa che evidentemente non ho chiarito a sufficienza nelle slide e/o a voce. Ci sono tre tipi di gioco, e tre classi di età. Il numero di combinazioni dunque è nove. Di queste nove, voi dovete farne tre. *Almeno* tre, ovviamente. Scegliete voi cosa fare e come metterle, ma una storia per ogni tipo di gioco e una storia per ogni tipo di età. Quindi ogni colonna e ogni riga della matrice tipo x età deve avere una casella occupata. Quale, lo scegliete voi.

**Q:** che differenza c'è tra classe e gruppo, si possono unire i due gruppi in uno non individuale?

**A:** Quando l'abbiamo pensato, abbiamo ipotizzato la visita di ragazzi ad un museo. Quindi i ragazzi sono da soli, oppure in piccoli gruppi (tre/quattro ragazzi, anche di età diverse), oppure una classe (20-25 ragazzi della stessa età e un chiaro fine educativo). La vera differenza secondo me è che una classe non può muoversi tutta in blocco per realizzare le esperienze proposte dalla storia, per cui è necessario spezzare la storia in sequenze separate ed indipendenti che vengono attribuite a piccoli gruppi. Qui si possono far sorgere cose come competizione tra gruppi, gare di velocità o precisione, ecc. che rendono il gioco più interessante.

**Q:** Sempre nella stessa slide c'è scritto "almeno 10 attività". in queste 10 rientrano quelle di tipo esplicativo?

**A:** E' necessario prevedere delle attività puramente passive e di lettura, in cui il ragazzo si immerge in una trama e in un contesto narrativo preciso (un'investigazione, uno spionaggio, un giallo, ecc.) e delle attività in cui affronta delle sfide (fisiche, visive o di ragionamento) con degli oggetti della vita vera che vengono fatti interagire con la storia. Quanti? Boh. Sicuramente sono sbagliate le storie senza ambientazione, e le storie composte solo da ambientazione. Quindi è necessario trovare un equilibrio. Almeno due attività di lettura (una iniziale e una finale) poi immagino che ci saranno delle fasi intermedie di snodo della trama da gestire. Le altre sono sfide.

**Q:** Tutte e tre le storie devono essere accessibili o ne basta solo una?

**A:** I requisiti di accessibilità (solo 9CFU) sono i seguenti:

1. *Almeno* una storia deve essere accessibile.
2. L'editor deve permettere all'autore della storia di fornire i contenuti per l'accessibilità
3. Ovviamente il player, per quanto di sua competenza e indipendentemente dalla storia caricata sopra, non deve fornire problemi di accessibilità propri.

**Q:** *Una volta che l'ambiente autore viene usato dal genitore per aggiungere interazioni tra la storia e l'ambiente, ci sono maniere particolari con le quali mostrare il QR Code? Basta semplicemente che il software esporti un'immagine del code e che i bambini lo inquadrino direttamente dal cellulare del genitore?*

**A:** Il QRcode è un'immagine generata automaticamente dal software autore. Essa può essere stampata oppure mostrata a schermo al ragazzo che inizi una nuova partita.

**Q:** *Ci sono vincoli riguardo le storie? Dobbiamo creare n storie diverse o possiamo riutilizzarne una per diversi tipi di gioco/fasce d'età? (Sarebbe in ogni caso nostra responsabilità il rendere la storia compatibile con gruppi di diverse dimensioni).*

**A:** Le tre fasce di età si differenziano tra loro non solo per la vastità e sofisticazione dei contenuti delle storie, ma anche e soprattutto per il tipo di sfide che debbono essere loro proposte, sulla base dei modelli di ragionamento di Piaget e Bruner. Se riuscite a creare storie diverse all'interno della stessa trama, questo è ottimo. Sono un po' più scettico all'idea di riciclare attività e teesti.